7Б

УДК 004.318

В.П. Бурдаев
Харьковский национальный экономический университет, г. Харьков, Украина
burdaevvp@mail.ru

Модель функциональной системы динамической предметной области

Рассматривается модель функциональной системы динамической предметной области, реализованная в системе «КАРКАС» на основе понятия расслоения.

Введение

Необходимость построения динамических моделей предметной области является одной из причин их использования как людьми, так и программными агентами. Например, консорциум W3C разрабатывает OWL (ontology web language), с помощью которого предметная область может быть представлена в виде модели объект-свойство для программных агентов, которые осуществляют поиск информации. В этом смысле онтологии представляют собой интеллектуальные средства для развития и совершенствования сети Интернет [1-3].

Модель онтологии предметной области в системе «КАРКАС» [4], [5] состоит из иерархии классов предметной области, связей между ними (правил вывода), которые действуют в рамках этой модели. В системе предложен механизм интерпретации онтологии в условиях динамического изменения ее параметров (базового класса, связей между классами и взаимодействия объектов классов).

Цель работы состоит в разработке онтологического подхода к представлению знаний предметной области, обладающих динамической структурой и возможностью самоадаптации в процессе эксплуатирования в компьютерной системе «КАРКАС».

Модель онтологии, основанная на понятии расслоения

Онтологии используются для представления знаний о понятиях предметной области (ПрО) и предположительных отношениях между ними. Почти все модели онтологий, содержат концепты (сущности, понятия, классы, объекты), свойства концептов (свойства, атрибуты, слоты), отношения между концептами (связи, зависимости).

Физическая модель базы знаний (БЗ) хранит экземпляры классов, объектов, значения атрибутов объектов и логические связи между классами, объектами.

Частично упорядоченность классов в онтологии строится путем выделения, так называемого базового (целевого) класса, располагающегося на самом верхнем уровне иерархии классов. Далее выделяются классы, находящиеся на следующем уровне, и так далее.

Функциональная система – это система, сформированная для достижения заданного полезного результата (целевой функции) в процессе своего функционирования. Следовательно, ее системообразующим фактором является конкретный результат функционирования системы. Другими словами, цель рассматривается как заданный результат, а ограничения – как степень свободы, необходимая для достижения результата.

Модель предметной области рассматривается как функциональная система, в которой результат оказывает организующее влияние на все этапы формирования онтологии. Классы и связи между ними можно рассматривать как логическую конструкцию функциональной системы.

Отличительной особенностью функциональных систем – в их открытости, неавтономности, неизолированности от внешней среды. Математической моделью, описывающей эволюцию таких систем, служат неавтономные дифференциальные уравнения.

В глобальной качественной теории автономных дифференциальных уравнений важную роль играет понятие структурной устойчивости, введенной Андроновым и Понтрягиным под названием грубости, которое означает, что фазовый портрет потока не меняется (с точностью до топологической орбитальной эквивалентности) при всех достаточно малых изменениях векторного поля.

Определение и изучение структурной устойчивости неавтономных дифференциальных уравнений рассмотрены в работе [6]. Приведем не вполне строгое определение этого понятия, ограничиваясь для простоты моделью:

$$\begin{cases} dx/dt = f(x,y) \\ dy/dt = g(y). \end{cases} \quad (1)$$

Скажем, что расширение (1) структурно устойчиво, если для всех достаточно малых возмущений $r(x,y)$ система

$$\begin{cases} dx/dt = f(x,y) + r(x,y) \\ dy/dt = g(y). \end{cases} \quad (2)$$

сопряжена системе (1), причем сопрягающий гомеоморфизм тождествен по второму аргументу, то есть имеет вид $h(x,y) = (h_1(x,y), y)$. Таким образом, возмущается только первое уравнение, описывающее объект, а второе уравнение, моделирующее среду, остается неизменным. Класс возмущений уже, чем при обычном определении структурной устойчивости системы (1), но зато на сопрягающий гомеоморфизм наложено дополнительное условие.

Приведем описание модели (1) в терминах глобального анализа. Тройка объектов (M, p, B) образует расслоение, где $p: M \to B$ – непрерывное отображение (проекция), B – база расслоения, $M_b \equiv p^{-1}(b)$ – слой расслоения.

Пусть (M, p, B) – гладкое расслоение и $X: M \to TM$ и $Y : B \to TB$ – два векторных поля, причем $Tp(X(m)) = Y(p(m))$ $(m \in M)$. В локальных координатах поле X задается системой вида (1), поле Y – вторым уравнением этой системы.

Пусть (M, R, π) и (B, R, ρ) – фазовые потоки векторных полей X и Y соответственно. Тогда отображение $p: M \to B$ является гомоморфизмом потока (M, R, π) на (B, R, ρ), т.е.

$$p(\pi^t(m)) = \rho^t(p(m)) \ (m \in M, t \in R).$$

В таком случае говорят, что поток (M, R, π) является расширением потока (B, R, ρ). Часто само отображение $p:(M, R, \pi) \to (B, R, \rho)$ расширением динамических систем. При этом база B интерпретируется как фазовое пространство среды, а фазовые состояния изучаемого объекта при фиксированном состоянии $b \in B$ внешней среды изображаются точками слоя $M_b \equiv p^{-1}(b)$.

Теорема о структурной устойчивости. Пусть M – компактное многообразие класса $C^{2,3}$ над B с проекцией $p : M \to B$ и векторное поле X принадлежит $X^{0,2}(M)$. Предположим, что векторное поле X удовлетворяет строгому условию трансверсальности в вертикальном направлении. Тогда расширение $p:(M, R, \pi) \to (B, R, \rho)$ $C^{0,1}$ – структурно устойчиво.

Пусть B – множество атрибутов предметной области, V – множество их значений, $p : B \times V \rightarrow B$ проекция, $V_b \equiv p^{-1}(b)$ – слой расслоения над точкой b, то есть множество значений атрибута b. В прямом произведении B x V определим отношение R, наделенное следующим свойством: для любого атрибута $b \in B$ существует значение $v \in V_b$ такое, что $R \equiv bRv$. Другими словами, отношение R определяет принадлежность значений из слоя V_b конкретному атрибуту, то есть задает факты ПрО.

Пусть Ф – множество фактов ПрО и $r : Ф \rightarrow Ф$ – правило (если <условие> то <действие>). Тогда объекты ПрО определяются как атрибуты, которые находятся в правой части правила и формируют факты только в том случае, когда правило выполняется (<условие> – истинно). Среди объектов выделяется целевой, который является системоорганизующим. Затем выделяется другие объекты, которые располагаются в определенной иерархии в соответствии с их правилами и могут быть отображены с помощью орграфа.

Представим себе, что система $F \in B \times V$ параметризована некоторым множеством N в том смысле, что каждому $t \in N$ соответствует некоторое состояние работы системы, которое отображается орграфом, как, например, на рис. 1. Интуитивно под структурной устойчивостью функциональной системы F будем понимать, что существует некоторая «окрестность» орграфа системы F, в которой орграфы «близких» к ней систем «топологически» эквиваленты исходному. На рис. 1 приведен пример структурно устойчивой функциональной системы.

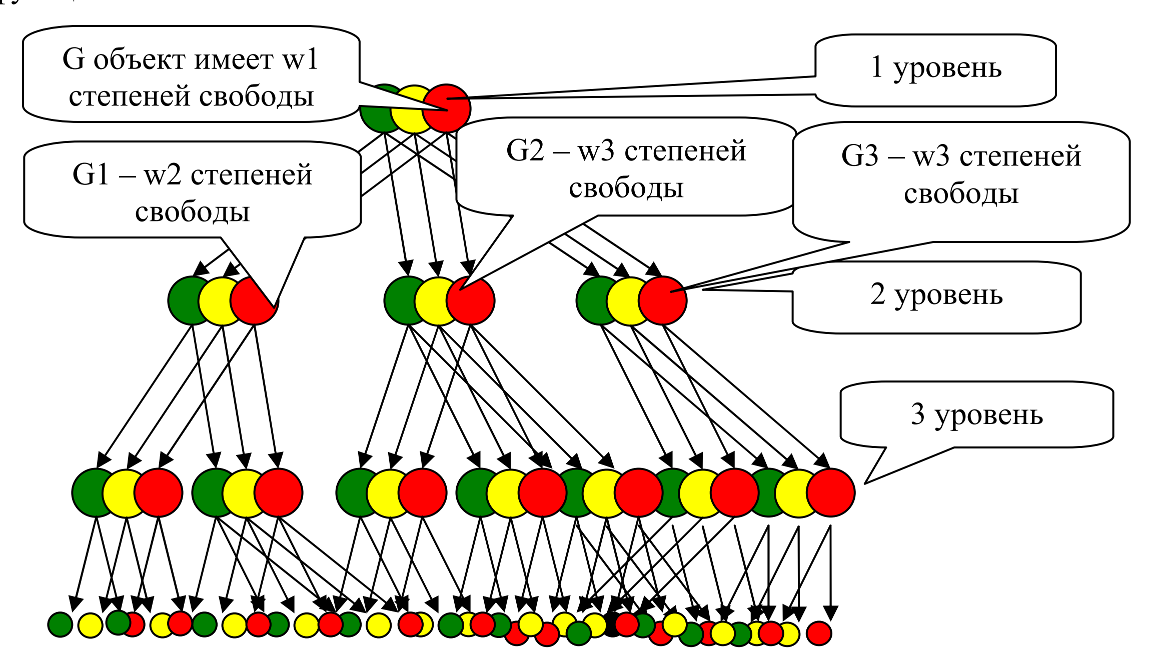


Рисунок 1 – Структурная устойчивость функциональной системы

В работе [5] рассмотрены различные модели построения онтологий. Например, при разработке онтологии по определению вида экспозиции при оценке экологического риска была построена иерархия объектов, представленная на рис. 2. Онтология ПрО содержит 5 классов и двухуровневую структуру классов. В БЗ осуществлена таксономия правил и фреймов, что позволило улучшить структуру онтологии.

Онтология в системе «КАРКАС» − это конструкция для агрегирования, множественных иерархий классов и правил вывода. В модели онтологии используются два класса, реализованные виде продукции и фрейма. Класс рассматриваются как шаблон, на основе которого создаются его экземпляры.

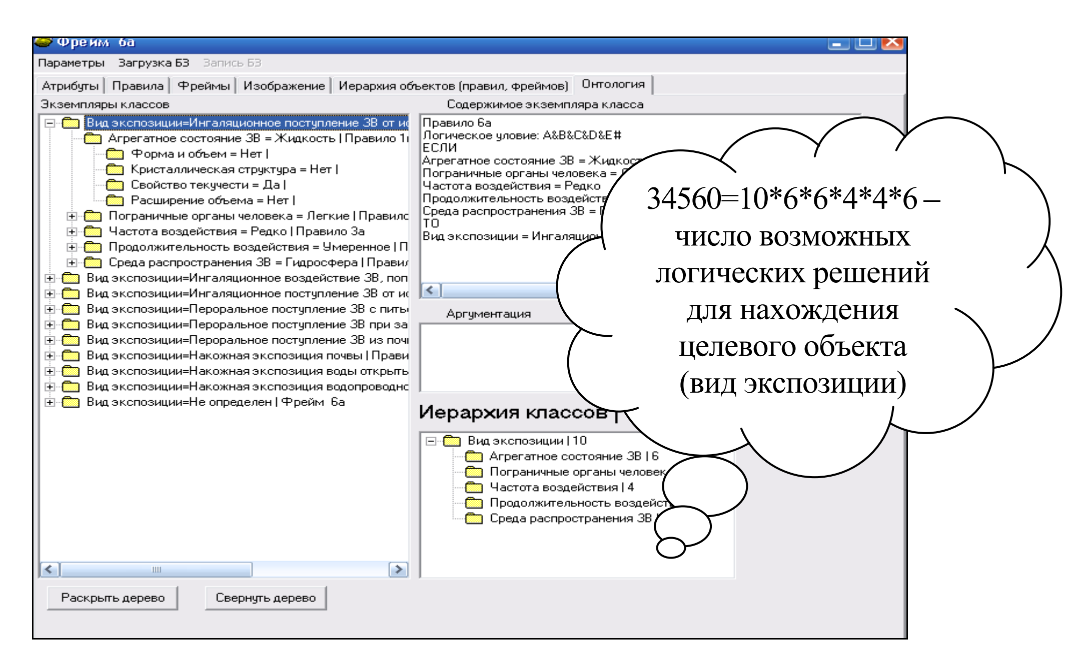


Рисунок 2 – Экземпляры класса для объекта «Вид экспозиции»

Разработка онтологий для «КАРКАС» состоит из 5 шагов:
определение границ онтологии;
определение доменов атрибутов ПрО;
определение классов, объектов с помощью продукций и фреймов;
построение орграфа, частично упорядоченность объектов и организация их иерархии;
формирование продукций и фреймов для описания классов, подклассов, экземпляров.

Модель формирования фактов онтологии в системе «КАРКАС»

Пусть $A = \{a_1, a_2,..., a_n\}$ – множество атрибутов предметной области, V_j – множество возможных значений $a_j \in A$ и $V = U_{a \in A} V_a$. Атрибуты могут быть измерены в разных шкалах (количественной, порядковой, качественной, смешанной).

Между атрибутом a_j и его значением v_j определим следующие операции:

1) $a_j = v_j$, = – знак операции означивания;

2) $a_j < v_j$, < – знак операции отношения (>, >=, <=);

3) $a_j \in [v_{j1} ,..., v_{jm}]$, $\in$ знак – операции принадлежности ([ ), ( ), ( ]).

Атрибут, его значение и операция между ними определяют высказывание. Пусть $Q= \{q: A \text{ -> } V\}$ множество высказываний.

На множестве Q определим две функции:

1. Функция $\mu : Q \to [0,\dots,1]$, позволяет интерпретировать высказывание q с точки зрения его истинности.

2. Функция $\theta : Q \to [0,\dots,1]$, позволяет интерпретировать высказывание q с точки зрения его значимости.

Тогда триплет $f = (q, \mu(q), \theta(q))$, где $q \in Q$, назовем фактом предметной области.

Пусть λ – отображение из прямого произведения $Q \text{ x } [0,\dots,1] \times [0,\dots,1]$ в $Q \times [0,\dots,1] \times [0,\dots,1]$. Через $\Gamma = \{ \lambda \}$ множество всех возможных отображений таких, что $\lambda(f) \in Q \times [0,\dots,1] \times [0,\dots,1]$.

Определим ситуацию s как набор фактов, связанных между собой знаками конъюнкции, дизъюнкции или отрицанием. Например,

$$s = f_1 \ \& \ f_2 \ U \ (\neg f_3) \qquad (3)$$

Замечание. Отрицание $\neg f_3$ применяется только для операции означивания и означает возможность альтернативного выполнения всех других событий из исчерпывающего множества (в смысле логической операции ИСКЛЮЧАЮЩЕЕ ИЛИ), например,

$$\neg f_3 = \neg (a = v_j) = (a = v_1) \ U \dots U \ (a = v_{j-1}) \ U \ (a = v_{j+1}) \ U \dots U \ (a = v_n)$$

$v_j \in V_a$, $j = 1, \dots, n$.

Обозначим через $S = \{s\}$ множество всех ситуаций предметной области и $p: S \to S$ некотрое преобразование множества S на себя. Рассмотрим итерации этого отображения, то есть результаты его многократного применения к точкам фазового пространства. Они задают динамическую систему с фазовым пространством S и множеством моментов времени N. Действительно, будем считать, что произвольная точка $s_0 \in S$ за время $t = 1$ переходит в точку $s_1 = p(s_0) \in S$. Тогда за время $t = 2$ эта точка перейдет в точку $s_2 = p(s_1) = p(p(s_0))$ и т.д.

Пусть $R = \{r: S \to S\}$ множество преобразований (правил) следующего вида:

r : ЕСЛИ <АНТЕЦЕДЕНТ> ТО <КОНСЕКВЕНТ>

где <АНТЕЦЕДЕНТ> (условие – ситуация s_1), <КОНСЕКВЕНТ> (вывод, следствие – ситуация s_2). Если ситуация s_1 в правиле принимает значение истина, тогда ситуация s_2 получает значение истина и добавляется к S. Другими словами, если ситуация $s_1 \in S$ в правиле принимает значение истина, тогда существует отображения q_j: A -> V формирующее ситуацию s_2

$s_2 = q_1 * q_2 \ \ * q_n$ и * знак операций & , U ,$\neg$.

Если ситуация s_1 в правиле принимает значение ложь, тогда ситуация s_2 не добавляется к S. Например, ЕСЛИ s ТО q_4, Здесь s – ситуация (3), q_4 – некоторое высказывание. Результатом выполнения правила будет создан новый факт:

$$f_4 = (q_4 , \mu(q_4), \theta(q_4)),$$

$$\mu(q_4) = k * \max [\min(\mu (q_1), \mu (q_2)), (1 - \mu (q_3))]. \qquad (4)$$

$$\theta(q_4) = k * \max [\min(\theta(q_1), \theta(q_2)), (1 - \theta(q_3))]. \qquad (5)$$

Алгоритм построения онтологии:

составление полного и непротиворечивого логического описания ПрО;

формирование множества высказываний Q = {q: A -> V}

формирование объектов на основе правил из множества $R = \{r: S \to S\}$

установка иерархии объектов (построение дерева логического принятия решения, таксономии объектов).

Системы правил и фреймов являются основным способом синтеза и представления множеств (планов) отношения на множестве объектов. Фактор, который упорядочивает объекты (частичный порядок) и превращает ее в целенаправленную систему, – это отображения r.

Классы составляют самоорганизующую совокупность для работы функциональной системы.

Замечание 1. Частичный порядок объектов осуществляется в результате, когда консеквент одного правила, например, p_1 содержится в антецеденте правила p_2, тогда, объект 2 старше объекта 1.

Машина вывода состоит из двух частей: первая часть – анализатор правил, а вторая – механизм, позволяющий наделить множество S определенной структурой. Например, самая простая структура – это линейная, позволяющая для определения некоторого факта или ситуации установить цепочку применений преобразований из S:

$$r_0 \to r_1 \to r_2 \ \to \dots r_{n-1} \ \to r_n .$$

Здесь r_0 – начальное правило, а r_n – результирующее правило, определяющее результат функционирования функциональной системы, а сама цепочка правил – траектория функциональной системы.

Для того чтобы машина вывода смогла выполнить самый элементарный шаг, она должна вначале активизировать анализатор правил, на вход которого подается текущее правило. Если правило применимо (значение антецедента истинно), то анализатор правил конструирует новый факт или новую ситуацию. В противном случае правило не рассматривается. В функции анализатора правил входит:

1. Выполнить синтаксический анализ антецедента правила.
2. Вычислить булевское значение антецедента правила.
3. Если значение антецедента истинно, то сформировать факт или ситуацию в зависимости от консеквентна правила.

Заметим, что консеквент правила обычно представляет собой одно высказывание. В случае если консеквент имеет несколько высказываний, то формула (3) применяется к каждому высказыванию и в результате получается простейшая ситуация – конъюнкция фактов.

Анализатор правил используется для проверки синтаксиса написания правил как в редакторе БЗ, так и в логической машине вывода системы «КАРКАС» для вычисления коэффициентов уверенности фактов [4], [5].

Важная роль атрибутов заключается в том, что они задают отношение между объектами онтологии, например, значением атрибута является другой объект. Таким образом, порождается связь, вложенность объектов друг в друга.

Техническая реализация онтологии

Для представления атрибутов (свойств) используется следующая модель: < N D K_1 K_2 A>. Здесь N – лексема атрибута. Элемент D – домен атрибута. Элемент K_1 – это значение функции $\mu : Q \to [0,\ldots,1]$, позволяющая интерпретировать высказывание q с точки зрения его истинности. Элемент K_2 характеризует значимость атрибута (функция $\theta : Q \to [0,\ldots,1]$). Элемент A характеризует аргументированность атрибута (дополнительная информация эксперта об атрибуте).

Для представления правил используется следующая модель: < N L S1->S2 K_1 K_2 A>. Здесь N – лексема правила. Элемент L – логическое условие правила (логическое выражение принимающее значение истинно или ложно). S1->S2 – ядро правила (S1- антецедент, S2 – консеквент). Элемент K_1 характеризует степень уверенности эксперта в истинности ситуации S2 . Элемент K_2 характеризует значимость правила. Элемент A характеризует аргументированность правила.

Для представления фреймов используется следующая модель: < N S G U K_1 K_2 A>. Здесь N – лексема фрейма. Элемент S – слот (атрибут). Слот может содержать не только конкретное значение атрибута, но также имя процедуры (сервиса), позволяющей вычислить это значение по заданному алгоритму. Сервисы автоматически запускаются при обращении к соответствующему слоту. Слоты – объекты или другие фреймы. Элемент G – целевой слот (вызывает сервис для формирования объекта). U – указатель наследования, который показывает, какую информацию об атрибутах активной базы знаний наследуют слоты с аналогичными атрибутами в данном фрейме. В системе «КАРКАС» символ «н» означает, что значение слота наследуется. Действия указателя наследования заключается в следующем, если в результате работы машины вывода образовался некоторый факт, то все фреймы, содержащие слоты с именем атрибута в факте и с сервисом ЗАМЕЩЕНИЕ,

получают значение атрибута из факта. Машина вывода использует и ряд других сервисов: СУММА, ВВОД, СУММА ЗНАЧЕНИЙ, ОПРЕДЕЛИТЬ. Элемент K_1 характеризует степень уверенности эксперта в истинности значения целевого слота. Элемент K_2 характеризует значимость фрейма. Элемент А характеризует аргументированность фрейма.

Машина вывода – программный сервис, моделирующий алгоритм рассуждений с целью получения новых фактов из БЗ для решения задачи. Машина вывода имеет несколько режимов, использующих различные методы вывода: обратный вывод (от гипотез к данным) установлен по умолчанию и реализован в виде рекурсивной процедуры; прямой вывод (от данных к гипотезам); байесовский вывод (применение формулы Байеса); нейлоровский вывод (применение алгоритма цены свидетельств).

«Доска объявлений» (модуль машины вывода системы «КАРКАС») служит как для регистрации промежуточных результатов, так и для управления логическим выводом.

Машина вывода реализует алгоритм упорядоченного подбора целей, анализа правил и взаимодействия с пользователем для нахождения основной цели консультации. В качестве цели может выступать любой объект. В начале консультации машина вывода снимает с «доски объявлений» активные объекты и создает, если это необходимо, заключительное правило и начинает формировать список гипотез для просмотра классов БЗ. При просмотре классов их объекты получают статус подцели. В соответствии со структурой класса машина вывода определяет иерархию рассуждений, приводящих к нахождению значения подцели. В случае если значение подцели не найдено, то пользователю предлагается ввести его.

Метазнания о БЗ представлены метаправилами, которые просматриваются машиной вывода в прямом направлении, то есть от образованных фактов, анализируемых с помощью «доски объявлений», к активизации нужных классов или выработки конкретных рекомендаций для дальнейшей работы машины вывода. Гипотезы формируются в результате разбора действия заключительного правила и заносятся на «доску объявлений». Эксперт имеет возможность изменить порядок формирования гипотез с помощью репрезентативных атрибутов. Заключительное правило преобразуется в соответствии со стратегией эксперта.

В архитектуре системы «КАРКАС» можно выделить следующие основные модули (агенты): загрузчик; визуальный редактор для разработки онтологии; модуль консультации; модуль кластеризации данных.

Загрузчик осуществляет запуск системы и координирует взаимодействие всех модулей. Визуальный редактор для разработки онтологии предоставляет методы доступа к знаниям. Для этой цели используется гибридное представление знаний, включающее правила и фреймы. Модуль консультации содержит следующие компоненты: машину вывода (выбор правил и фреймов из онтологии), блок объяснения, анализатор тестов, доску объявления, монитор базы фактов. Модуль кластеризации данных позволяет осуществлять интерактивную и интеллектуальную классификацию многомерных данных. Для тестирования знаний используются генератор тестов и монитор преподавателя.

Модули системы представлены в виде программных пассивных агентов (взаимодействие их осуществляется с помощью массивов параметров и через общую память компьютера). Среда их взаимодействия – это система «КАРКАС».

Такие компоненты, как визуальный редактор БЗ, машина логического вывода, блок объяснения, являются общими для перечисленных модулей и выступают агентами для функционирования модулей.

В системе выделены два режима использования: режим когнитолога – специалист, моделирующий онтологию ПрО и конструирующий БЗ и режим проблемного специалиста, другими словами квалификация пользователя недостаточно высока, и поэтому он нуждается в помощи и поддержке своей деятельности со стороны системы.

Функции, реализованные в системе, представлены в табл. 1.

Таблица 1 – Функции системы «КАРКАС»

№	Описание функции
1.	Построение онтологии ПрО: классов, их экземпляров (объектов), атрибутов
2.	Частично упорядоченность классов и их визуализация в виде дерева объектов
3.	Использование технологи Microsoft Agent (агенты merlin, peedy и другие)
4.	Применение речевого интерфейса, анимированных персонажей для объяснения пользователю о том, как и почему сформирован факт во время консультации.
5.	Создание метаправил с помощью доски объявления.

В режиме обучения и тестирования применяются следующие функции:

импорт и экспорт вопросов и ответов для создания тестов (типы файлов: txt, xsl);

использование визуальных вопросов (пользователь может выбрать ответы на изображении);

настройка параметров тестирования;

визуализация оценок тестирования во время консультации;

анализ статистики тестирования.

Визуальный редактор БЗ – это программный модуль, предоставляющий когнитологу возможность создавать БЗ в интерактивном режиме. Редактор включает в себя шаблоны языка представления знаний (продукции, фреймы), подсказки и другие сервисные средства, облегчающие работу с базой.

Редактор БЗ помогает эксперту или инженеру по знаниям легко модифицировать и проверять БЗ.

Редактор БЗ содержит синтаксический контроль логического условия для продукции.

Для синтаксического контроля ввода атрибутов ПрО и их значений в антенцедентах продукции в помощь пользователю предложен механизм выбора их из выпадающего списка.

Визуальный редактор БЗ позволяет клонировать продукции. Создается правило для клонирования. Например, антенцедент продукции содержит четыре атрибута, число вариантов для первого атрибута равно 3, соответственно для второго – 5, для третьего – 4, для четвертого – 2 и консеквент имеет один объект. Тогда в результате клонирования этого правила система создаст 120 вариантов различных комбинаций правил. Когнитологу останется только удалить не нужные правила.

В системе «КАРКАС» атрибут имеет несколько вариантов ответов, при этом каждый ответ имеет коэффициент фактора уверенности, который проставляется экспертом в диапазоне [0,…,1] или в случае группы экспертов определяются методами экспертных оценок.

В системе имеется возможность выставить два коэффициента достоверности ответа, что позволяет использовать формулу Байеса для формирования априорных вероятностей, которые применяются при построении базы знаний, использующих вывод, основанный на методе Байеса. Применение коэффициентов фактора уверенности ответов позволяет при тестировании использовать математический аппарат нечетких множеств для адекватного отображения знаний, тестируемого на шкалу оценок.

В системе «КАРКАС» используются два класса, реализованные виде продукции и фрейма. Класс − это шаблон, на основе которого создаются экземпляры класса. Таким образом, онтологию можно также представить как совокупность взаимодействующих объектов.

Агент объяснения – это программный модуль, поясняющий ход рассуждения и принимаемых решений машиной вывода. Агент позволяет пользователю (когнитологу) получить ответы на вопросы: «Как было получено то или иное решение?». Алгоритм основан на трассировке машины вывода. Запросы к агенту возможны следующие: как образован факт, почему сработало правило. Агент разворачивает дерево принятия решения. Агент доступен во время консультации.

Агент реализует принцип ретроспективного рассуждения – он объясняет, как машина вывода достигла конкретного состояния БФ. Например, пользователь может пожелать узнать, почему машина вывода нуждается в ответе на вопрос, только что заданный. Или, как и почему образован факт. Агент может вывести вопрос с вариантами ответов или выполнить вывод и проследить цепочку правил и фреймов, приводящих к факту.

Агент может выполнить гипотетические рассуждения: пользователь вводит или изменяет факт, и машина вывода предъявляет, что может быть получено в БФ в таком случае.

Использование агентов дает возможность более точно моделировать диалоги в процессах консультации, обучения, тестирования, отладке БЗ.

Средства отладки БЗ. Трассировка машины вывода предоставляет пользователю следить за ее действиями: в режиме просмотра БФ выводится последовательные шаги консультации, имена правил и фреймов, которые были активизированы. Имеется возможность откорректировать факт, просмотреть варианты ответов на заданный вопрос, проследить причинно-следственную связь, вызов правил и фреймов.

Режим отладки консультации предоставляет возможность просмотреть дополнительные параметры обработки правил и фреймов, что обеспечивает систему прерываний консультации для анализа работы машины вывода.

Предоставляется возможность выбора цели (объекта онтологии) консультации для локального тестирования части онтологии предметной области.

Режим автоматического тестирования БЗ позволяет обнаружить ошибки, несовместимость решения задачи. Для пользователя эти возможности полезны в тех случаях, когда БЗ проверяется на полноту и непротиворечивость, когда БЗ модифицируется (пополняется или пересматривается). Другими словами, система осуществляет контроль над изменением БЗ с позиции поиска каких-либо ошибок.

Режим таксономии (фильтрации) БЗ позволяет разбить онтологию на классы и установить фильтрацию классов.

Замечание. Фильтрация классов важна для обратного вывода. Порядок расположения правил в классах значения не имеет.

Если машина вывода вовремя консультации не может найти информацию (атрибут, объект, правило, фрейм), то она запрашивает ее у пользователя, при этом указывается причина.

Консультационный режим. Консультационный режим – это интерактивный режим эксплуатации системы, при котором пользователь продвигается к решению задачи. Модуль, реализующий этот режим, обеспечивает ведение диалога с пользователем, в ходе которого система запрашивает у пользователя необходимые факты для процесса рассуждения; предоставляет возможность пользователю в определенной степени контролировать и корректировать ход рассуждений.

Алгоритм создания БЗ на основе системы «КАРКАС» состоит из следующих шагов:

1. Построить онтологию ПрО:

определить классы, объекты и их связи;

определить атрибуты (свойства);

определить цели и подцели;

определить критерии оценки (составить список);

составить список значений, которые может принимать каждый из определенных атрибутов;

выбрать способ определения степени уверенности.

2. Выполнить идентификации ПрО.
3. Построить концептуальную модель ПрО.
4. Выбрать логический вывод на основе проведенного выше анализа.
5. Выполнить формализацию БЗ.
6. Протестировать БЗ.

Выводы. С помощью системы «КАРКАС» разработан ряд прототипов ЭС в следующих предметных областях: медицина, экономика, мобильная связь и кластерный анализ многомерных данных [5], [6].

Литература

1. Палагин А.В. Системно-онтологический анализ предметной области. / А.В. Палагин, Н.Г. Петренко // Искусственный интеллект. – 2010. – № 3. – С. 608-617.
2. Клещев А.С. Математические модели онтологий предметных областей. Часть 1. Существующие подходы к определению понятия «онтология» / А.С. Клещев, И.Л. Артемьева // Научно-техническая информация. – 2001. – № 2. – С. 20-27. – (Серия 2 «Информационные процессы и системы»).
3. Осипов Г.С. Динамические интеллектуальные системы / Г.С. Осипов // Искусственный интеллект и принятие решений. – 2008. – № 1. – С. 47-54.
4. Бурдаєв В.П. Системи навчання з елементами штучного інтелекту / В.П. Бурдаев. – Харків : Вид. ХНЕУ, 2009. – 400 с.
5. Бурдаєв В.П. Моделі баз знань. / Бурдаев В.П. – Харків : Вид. ХНЕУ, 2010. – 300 с.
6. Бронштейн И.У. Неавтономные динамические системы / Бронштейн И.У. – Кишинев : Штиинца,1984. – 290 с.

Literatura

1. Palagin A.V. Iskusstvennyj intellekt. № 3. 2010. S. 608-617.
2. Kleshhev A.S. Matematicheskie modeli ontologij predmetnyh oblastej. Chast' 1. Sushhestvujushhie podhody k opredeleniju ponjatija “ontologija”. Nauchno-tehnicheskaja informacija, serija 2 “Informacionnye processy i sistemy”. № 2. 2001. S. 20-27.
3. Osipov G.S. Iskusstvennyj intellekt i prinjatie reshenij. № 1. 2008. S. 47-54.
4. Burdaev V.P. Systemy navchannja z elementamy shtuchnogo intelektu. Harkiv: Vyd. HNEU. 2009. 400 s.
5. Burdaev V. P. Modeli baz znan'. Harkiv: Vyd. HNEU. 2010. 300 s.
6. Bronshtejn I. U. Neavtonomnye dinamicheskie sistemy. Kishinev: Shtiinca.1984. 290 s.

В.П. Бурдаєв

Модель функціональної системи динамічної предметної області
Розглядається модель функціональної системи динамічної предметної області, реалізована в системі «КАРКАС» на основі поняття розшарування.

V.P. Burdayev

Model of Functional System of Dynamic Subject Domain
The model of functional system of dynamic subject domain, which is realized in the system «KARKAS» on the basis of concept of stratification, is considered.

Статья поступила в редакцию 16.06.2011.